

Tutorial 10: I/O System

Problem 1: Polling and Interrupt

Assume for a processor with a 450-MHz clock it takes 450 clock cycles for a polling operation (call polling routine, accessing the device, and returning). The overhead for an interrupt operation is 525 clock cycles. Hard disk transfers data in 16-byte, chunks and can transfer at 12 MB/second rate.

A . If the processor uses interrupt technique and the interrupt rate is equal to the software polling rate, what percentage of the processor time is tied up in servicing the interrupt by the hard disk during the data transfer?

B. If the processor's clock frequency increases to 475 - MHz and the processor uses interrupt technique and the hard disk is only active 2.5% of the time, what percentage of the processor time is tied up in servicing the interrupt by the hard disk?

C. If the processor's clock frequency increases to 800 - MHz and the processor uses interrupt technique and the hard disk is only active 3% of the time, what percentage of the processor time is tied up in servicing the interrupt by the hard disk?

D. To reduce interrupt overheads, the hard disk controller is augmented with a 64-byte buffer, and now only interrupts the CPU when the buffer is full. The interrupt rate is equal to the software polling rate. What percentage of the processor (450-MHz) time is tied up in servicing the interrupt by the hard disk during the data transfer?

D. The required interrupt rate can be computed as:

$$(\text{Interrupt Rate}) = (12 \text{ MB/sec}) / (64 \text{ Byte}) = (0.1875 \text{ Mega interrupts per second})$$

$$(\text{Number of clock cycles per second required for servicing interrupt}) = (0.75 \text{ Mega interrupts per second}) \times (525 \text{ clock cycles}) = (98.4375 \text{ Mega clock cycles per seconds})$$

$$(\text{Percentage loading due to polling}) = (393.75 \text{ Mega clock cycles per seconds}) / (450\text{MHz Processor Clock Frequency}) = 0.21875 = 22\%.$$

Problem 2: Handling Interrupts

Consider ARM DSLMU interrupt controller shown in Figure 1. Write an interrupt service routine that implements a re-entrant interrupt system with 8 priority levels (Input 0 with lowest priority and 7 with highest priority).

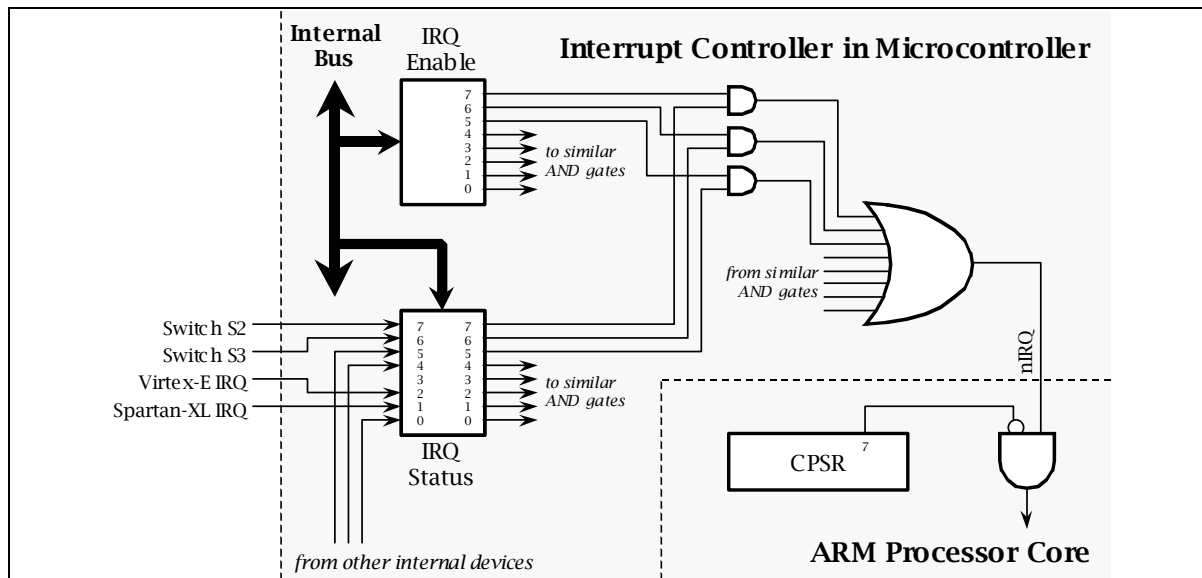


Figure 1: Interrupt Controller on the DSLMU Microcontroller Board

